BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di PT. Global Loyalty Indonesia (GLI) ini menjabat sebagai *Fullstack Application Development* dengan pengawasan dari Bapak Irfan Ahmad selaku *IT Development Manager*. *Team Developer* sebenarnya terdiri dari *Backend Application Developer* dan *Frontend Application Developer*.

Selain mendapat pengawasan dari Bapak Irfan Ahmad, beliau juga memberikan *outline* atau gambaran besar dari *project* yang akan dikerjakan kedepannya seperti fitur-fitur yang akan dibutuhkan dan memberikan *detail* pembangunan dari aplikasi yang akan dibuat.

Pada proses pengerjaan aplikasi diperlukan koordinasi dari senior *Backend Application Developer* dan *Frontend Application Developer* dikarenakan sebelum untuk menerjunkan aplikasi yang sudah dibuat harus diuji terlebih dahulu oleh senior dan sebelum diajukan kepada Bapak Irfan Ahmad dan senior memberikan solusi-solusi jika terjadi permasalahan dalam proses pembuatan aplikasi.

Selama pelaksanaan kerja magang tim dari *backend application developer* dan *frontend application developer* akan melakukan *meeting* melalui *google meet* dimana biasanya dilakukan saat awal-awal pemberian *project* yang diberikan dari Bapak Irfan Ahmad yang biasanya *meeting* tersebut membahas pekerjaan yang akan dikerjakan atau *progress* pembuatan dari aplikasi.

3.2 Tugas Yang Dilakukan

Selama aktivitas pelaksanaan kerja magang dilakukan, tugas yang diberikan dari Bapak Irfan Ahmad ialah membuat *backend* dan *frontend* aplikasi yang dikerjakan dimana aplikasi tersebut berbasis *web-application*. Pekerjaan yang diberikan menggunakan framework *codeigniter* dan *python flaskrestful*.

Dalam pembuatan aplikasi diawasi oleh Bapak Irfan Ahmad dan senior *backend application developer* dan *frontend application developer*.

3.3 Uraian Pelaksanaan

Praktik kerja magang dilaksanakan selama kurang lebih 19 minggu dengan rincian pekerjaan seperti di Tabel 3.5.

Tabel 3.5 Tabel pelaksanaan kerja magang

MINGGU KE	JENIS PEKERJAAN YANG DILAKUKAN MAHASISWA
1	• Perkenalan di tim IT Development • Meeting dengan HR dan Bapak Irfan
2	• Mempelajari codeigniter
3	• Mempelajari codeigniter • Mempelajari basic-basic python
4	• Mempelajari apa itu postman, dbweaver • Mempelajari menggunakan Flask
5	• Mempelajari menggunakan flask restful • Membuat basic CRUD python
6	• Mempelajari postgreSQL • Membuat function dengan method get • Meeting weekly dengan team
7	• Testing function method get menggunakan postman • Membetulkan function method get menggunakan postman
8	• Membuat function dengan method post
9	• Testing function menggunakan postman • meeting dengan team (penambahan pada method get)
10	• Memperbaiki error pada function post
11	• Testing ulang function method post • Menambahkan codingan di function method get

Tabel 3.5 Tabel pelaksanaan kerja magang (lanjutan)

MINGGU KE	JENIS PEKERJAAN YANG DILAKUKAN MAHASISWA
12	• Testing ulang function method get • Memperbaiki error function method get
13	• Testing ulang function method get • Membuat sketch mockup UI
14	• Membuat button post • Membuat tampilan form inject,login • Merging frontend dengan backend • Membuat format layout template backend
15	• Testing aplikasi dari login sampai inject Star • Menyelesaikan error pada aplikasi • Membuat frontend history
16	• Mengubah backend dari postgresql ke oracle • Membuat form history • Membuat backend history
17	• Testing form history • Memperbaiki bug
18	• Penambahan pada tampilan
19	• Deployment

Proses Pelaksanaan kerja magang terbagi menjadi beberapa bagian yaitu proses pengerjaan, mencari solusi pada *error* yang terjadi dan menerjunkan aplikasi yang telah dikerjakan.

3.3.1 Proses pelaksanaan

Selama dalam masa pembuatan aplikasi alfastar pada PT. Global Loyalty Indonesia, dipergunakannya sebuah *hardware* dan *software* untuk membuat program. *Software* yang digunakan antara lain:

1.Windows 10(64-bit)
2. Microsoft Visual Studio Code
3. DBeaver
4. XAMPP
5. Postman
6. Google Chrome
7. Github

Hardware yang digunakan untuk membuat aplikasi alfastar antara lain:

1. Processor laptop Intel Core i3 - 3217U GHz
2. RAM laptop: 4GB
3. Storage laptop:500GB

A. User Requirement

Berdasarkan arahan dari *supervisor* pembuatan aplikasi alfastar yang berbasis web, berikut adalah *requirement*:

1. Dapat Memasukan data *purchasing*
2. Dapat melakukan validasi data sudah ada di dalam database
3. Dapat *login*
4. Dapat menampilkan data history tabel *purchasing*
5. Dapat *logout*

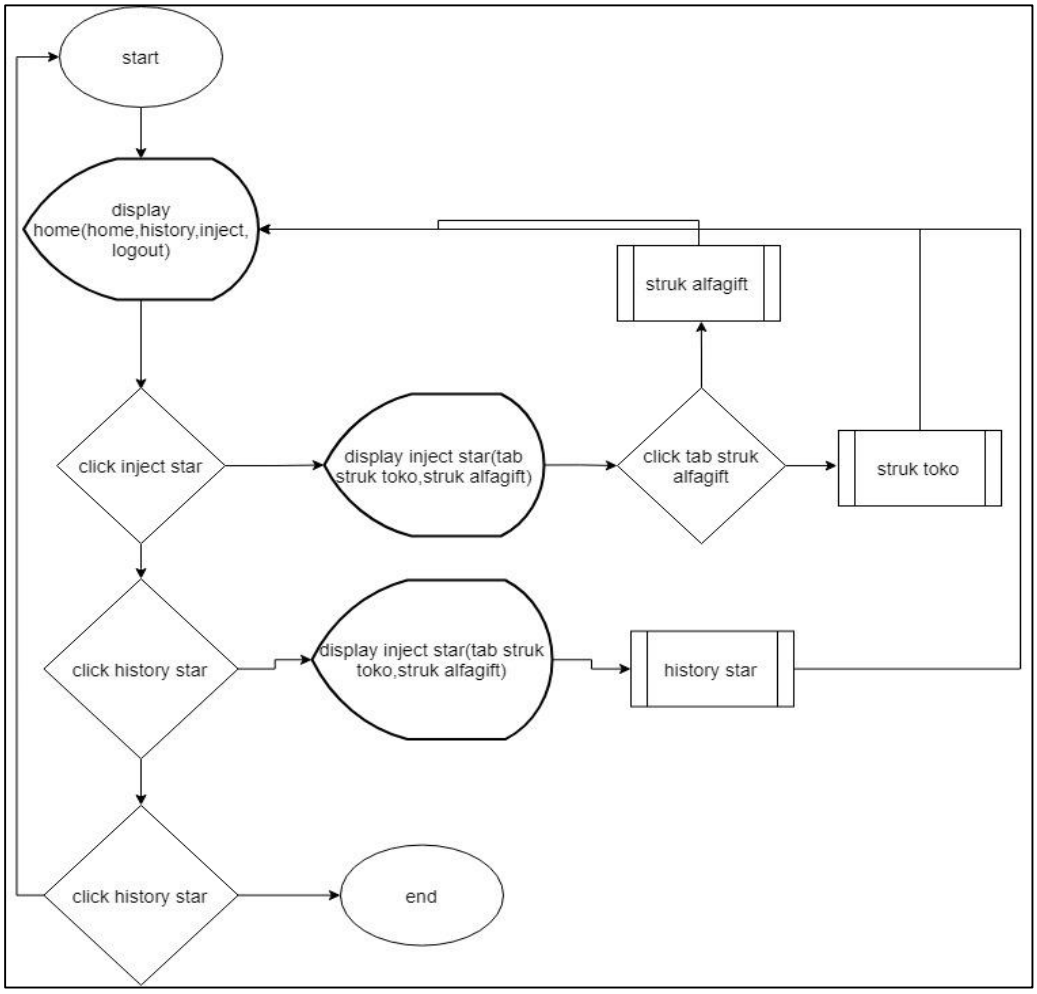
B. Perancangan Aplikasi

Berdasarkan arahan di atas dari *supervisor* pembuatan sistem aplikasi web alfastar dibuat seperti *user requirement*.

B.1. Flowchart

Berikut adalah *flowchart* diagram aplikasi alfastar atau alur dari aplikasi alfastar.

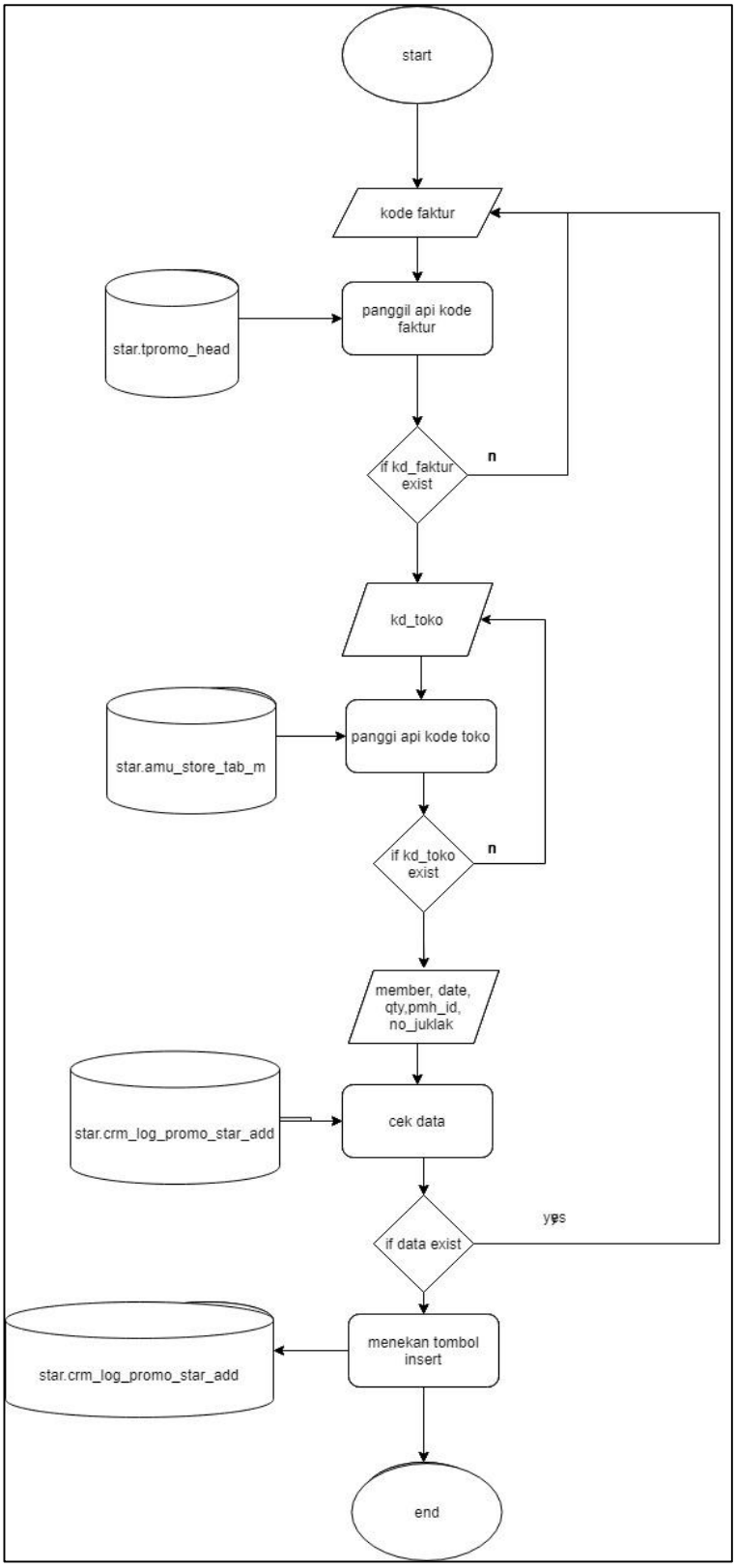
B.1.1 Flowchart Menu



Gambar 3.1 Flowchart home menu

Gambar 3.1 merupakan *flowchart* dari *home menu*. *Flowchart home menu* ini merupakan diagram yang menggambarkan alur user saat berpindah-pindah tab halaman ke tab halaman yang tersedia di *home menu* dimana *home menu* terdapat 4 tab halaman yaitu *home menu* itu sendiri, *inject star*, *history star* dan *logout*.

B.1.2 Flowchart Struk Toko

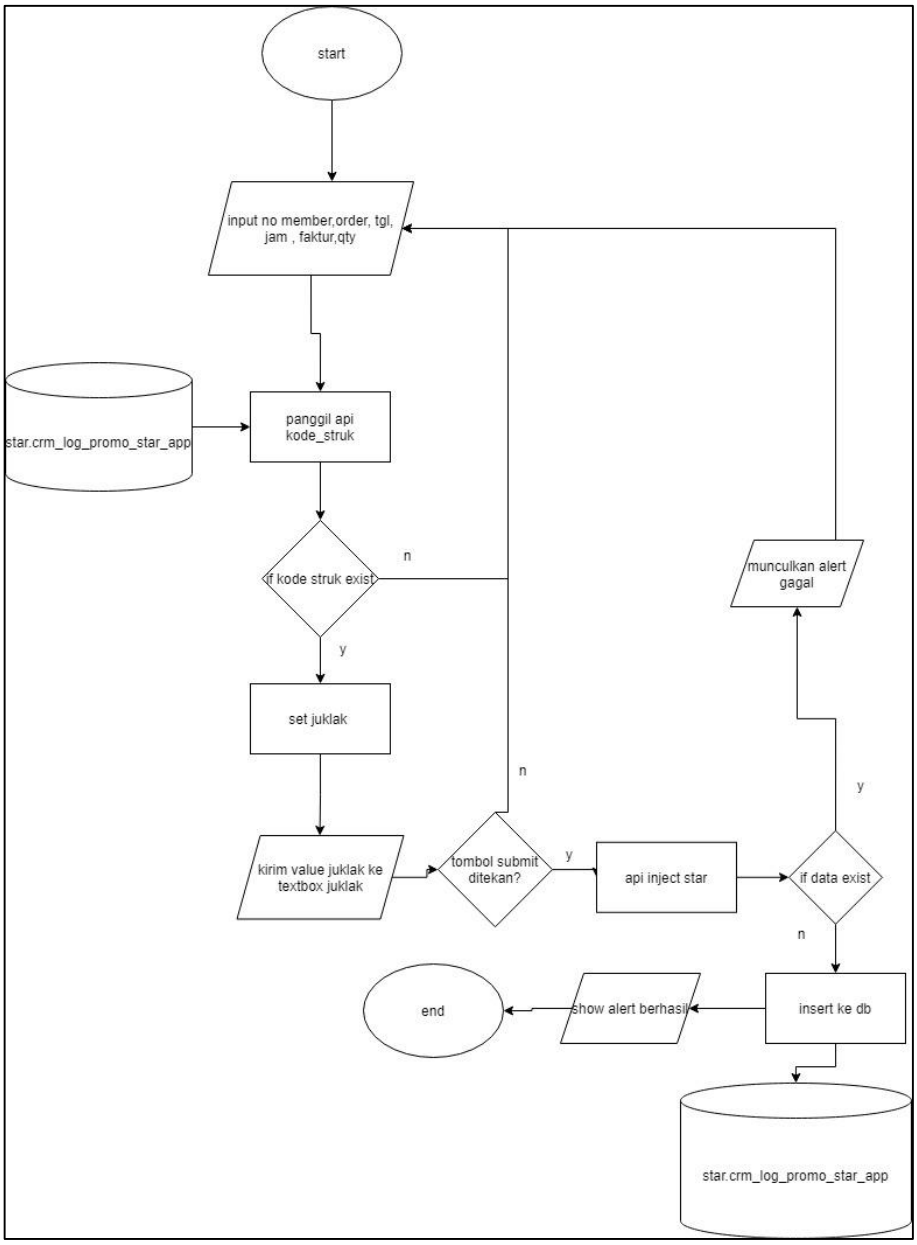


Gambar 3.2 Flowchart struk toko

Gambar 3.2 merupakan *flowchart diagram* struk toko dimana pada halaman Inject Star terdapat form struk toko yang berguna untuk membedakan *inject* Star dari toko atau bukan. Diagram ini berawal dari user memasukkan kode faktur, saat selesai memasukkan kode faktur maka akan memanggil api kode faktur menggunakan metode *get* yang bertujuan kode toko tersebut sudah ada atau belum di tabel star.tpromo_head. Jika kode faktur ada maka tombol *submit* pada form akan bisa ditekan akan tetapi jika tidak ada maka akan memunculkan *alert* yang berisi "Faktur tidak ditemukan" dan tombol *submit* akan dinonaktifkan dan user harus memasukkan ulang kode faktur.

Setelah itu user memasukkan kode toko maka akan memanggil api kode toko menggunakan metode get untuk memastikan data ada atau tidak dan sama seperti kode faktur kondisinya. Lalu user memasukkan data-data ke *field* yang harus diisi di form dan jika salah satu tidak diisi maka tombol *submit* akan dinonaktifkan. Setelah semua diisi user akan menekan tombol *submit* saat menekan tombol *submit button* akan memanggil API *inject* star dimana akan menjalani beberapa *query* yaitu *select* star.crm_log_promo_add dimana berfungsi memanggil data yang sudah ada dan dicocokkan ke *data* yang telah dimasukkan oleh *user* jika tidak ada maka akan lanjut menjalani *query insert* dan memasukan ke dalam tabel crm_log_promo_add di *schema* star. Jika isi dari *user* sudah ada di dalam tabel maka akan memunculkan alert "data sudah ada. Mohon diinput ulang" dan semua *value field* di dalam form akan dikosongkan.

B.1.3 Flowchart Struk Alfagift

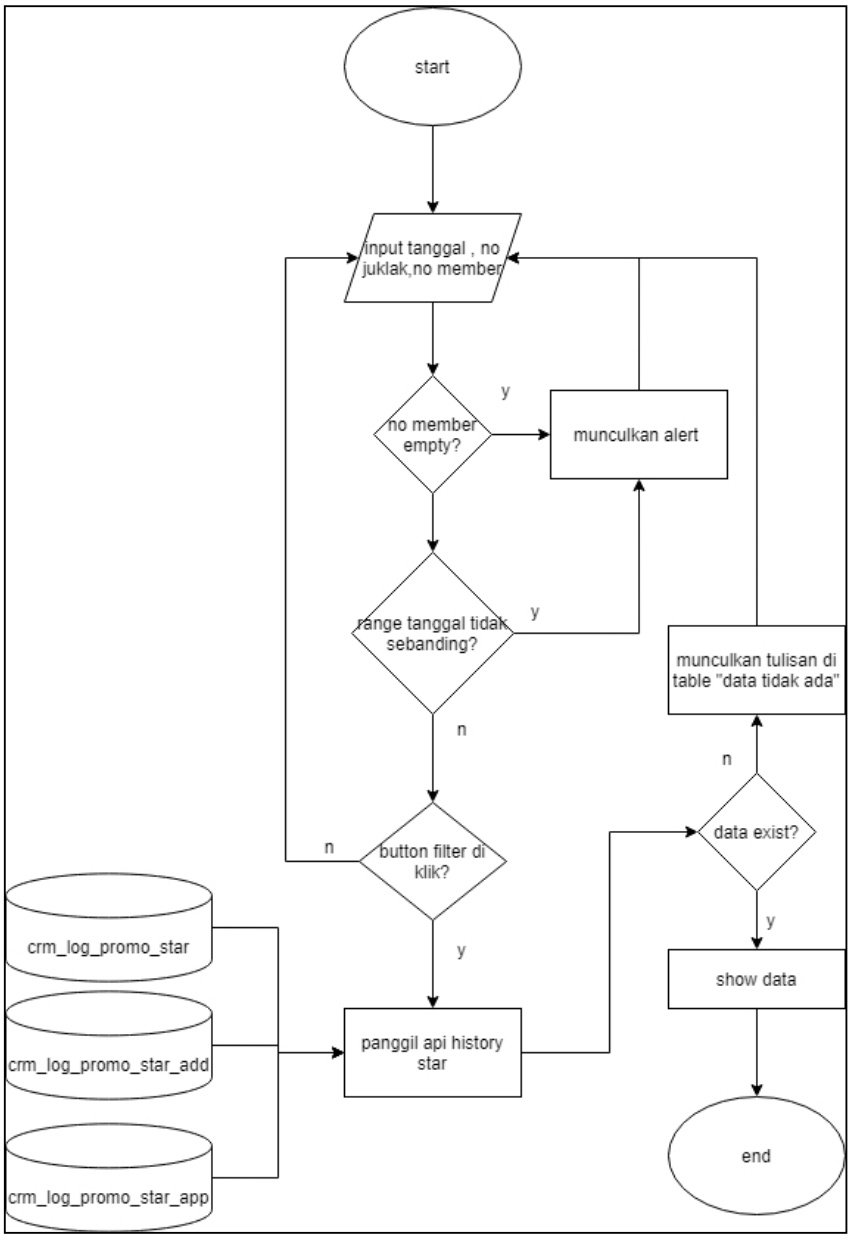


Gambar 3.3 Flowchart struk alfagift

Seperti Gambar 3.3 adalah *Flowchart* struk alfagift dimana berawal dari memasukkan data form yang berupa no member, no order, tanggal, jam transaksi, faktur dan *quantity*. Saat setelah memasukkan struk maka akan memanggil api kode struk yang berguna untuk melakukan pengecekan di dalam tabel crm_log_promo_star_add ada atau tidaknya kode struk yang dimasukkan, jika tidak ada maka *user* harus mengisi ulang kode struk. Jika kode struk ada maka *user* bisa

menekan tombol *submit*. Saat menekan tombol *submit* maka sistem akan memanggil API inject star dan melakukan pengecekan data yang tadi sudah diisi sudah ada atau belum. Jika sudah ada maka akan memunculkan alert " data sudah dimasukan ada" dan user harus mengisi ulang form. Jika data belum ada maka akan masuk ke dalam *database* dan memunculkan alert berhasil.

B.1.4 Flowchart History Star



Gambar 3.4 Flowchart history star

Pada Gambar 3.4 diagram menggambarkan tentang bagaimana proses *history* star terjadi. *User* harus memasukkan tanggal, no juklak dan nomor member. Jika no member tidak diisi akan memunculkan *alert* dan jika *range* tanggal tidak sebanding maka akan juga memunculkan alert. Jika semua sudah diisi maka user bisa menekan *button filter* yang bertujuan untuk memunculkan *history* yang dimana akan memanggil API *history* star dari tiga tabel yang berbeda yaitu crm_log_promo_star, crm_log_promo_star_add dan crm_log_promo_star_app. Jika data tidak ada maka akan memunculkan alert dari hasil *backend* sedangkan jika berhasil akan memunculkan hasil data yang berhasil diambil dari ketiga tabel tersebut.

B.2. Struktur API

Dalam melakukan proses aplikasi untuk memasukkan data dan mengambil data dari database dibutuhkannya API. Berikut adalah struktur API.

B.2.1 Struktur Tabel Star Add ,Star dan Star App

Tabel 3.6 Struktur tabel star add dan star

Variabel	Tipe Data
hari	date
faktur	varchar
store	varchar
nomor_faktur	varchar
nomor_member	varchar
quantity	int
pmh_id	int
nomor_juklak	varchar
date_create	time stamp
jam_transaksi	time stamp

Pada Tabel 3.6 merupakan struktur tabel star_add, star app dan star yang digunakan saat penambahan data atau inject star toko, inject star alfa gift dan

pemanggilan *history* yang menggunakan *method get* dan *post* .

B2.2 Struktur Kode Toko

Tabel 3.7 Struktur tabel amu store

Variabel	Tipe Data
kode toko	varchar
nama toko	varchar

Pada Tabel 3.7 adalah struktur tabel yang akan dipanggil ketika mengisi kode toko saat di form inject star toko dan menggunakan method get.

B2.3 Struktur Juklak

Tabel 3.8 Struktur tabel tpromo

Variabel	Tipe Data
faktur	varchar
nomor juklak	varchar

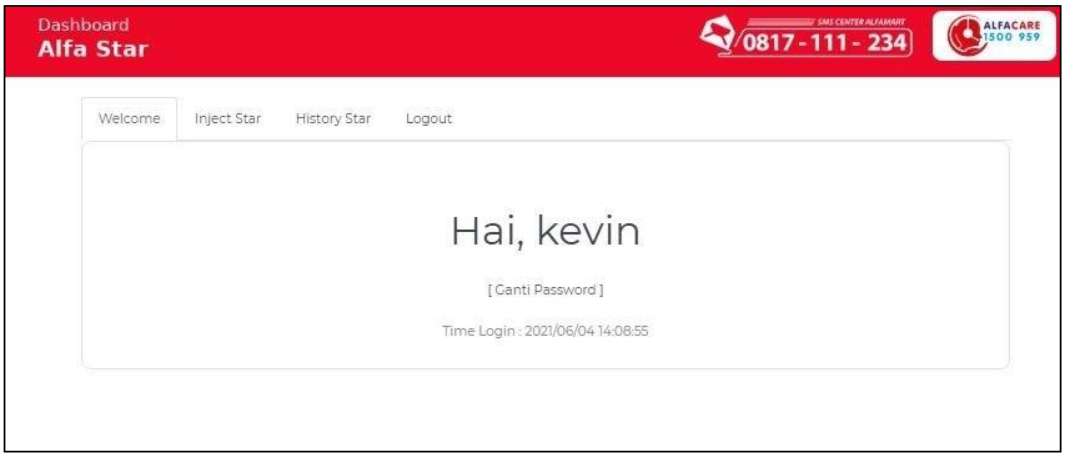
Pada Tabel 3.8 merupakan struktur yang digunakan untuk memanggil faktur dan nomor juklak yang dimana dipakai di form inject star kode toko dan inject star alfa gift yang menggunakan *method get*.

B.3 Rancangan User Interface

Berikut adalah rancangan antarmuka atau *user inteface* aplikasi alfastar yang sudah dibuat

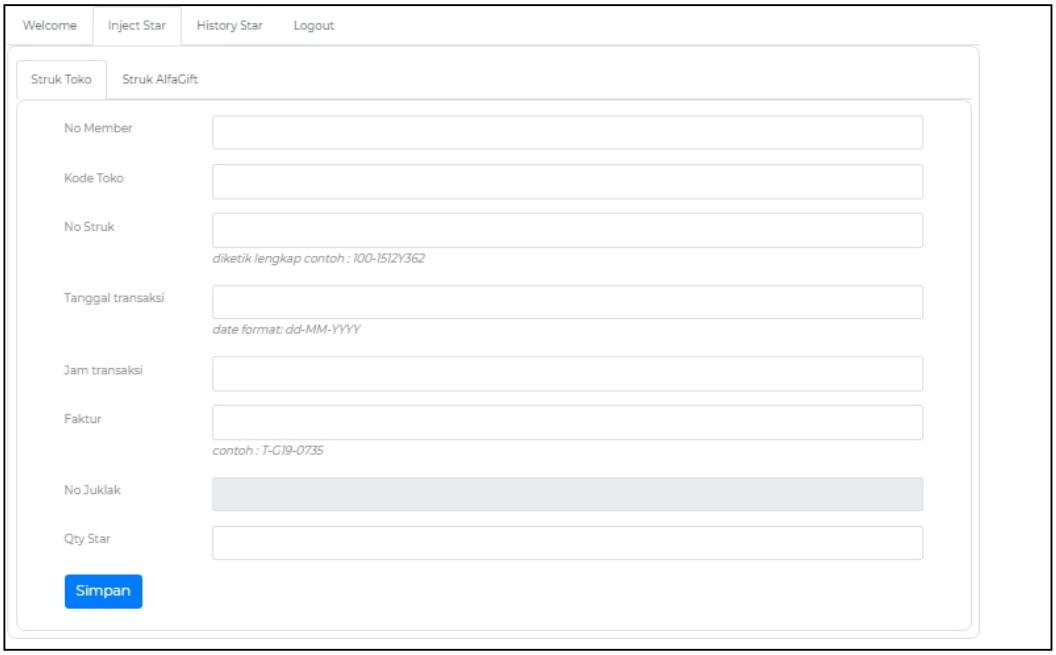
B.3.1 Halaman Home Screen

Pada Gambar 3.5 bisa dilihat tampilan halaman *home screen* saat memulai aplikasi web alfastar yang menunjukan bisa memilih tab inject star, history star dan bisa *logout* dari aplikasi.



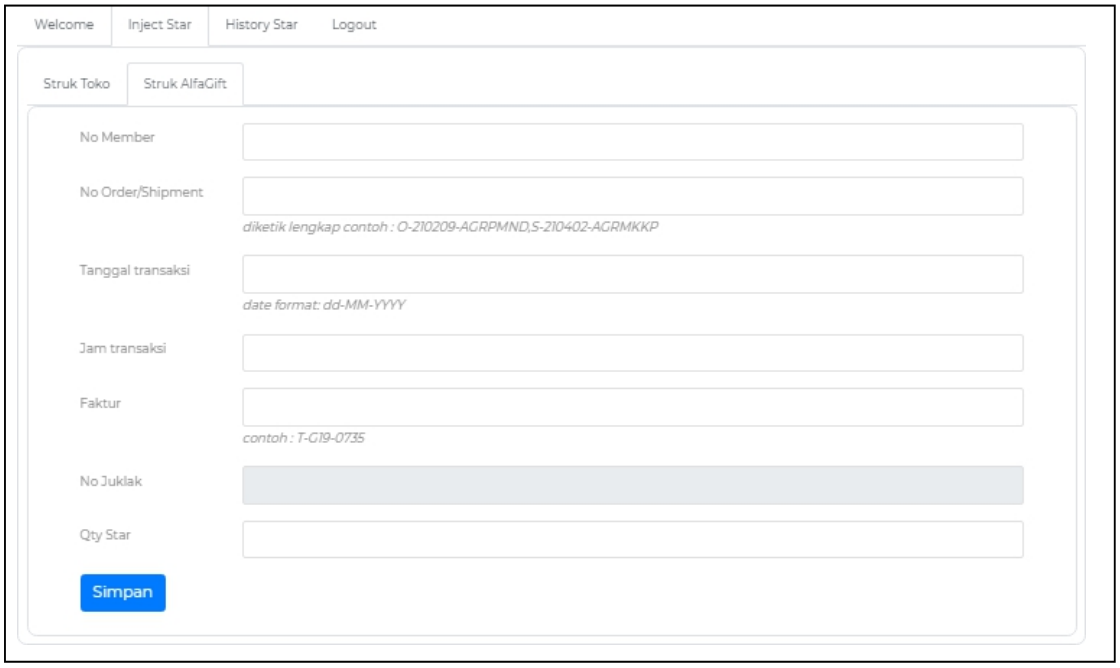
Gambar 3.5 Halaman home screen

B.3.2 Tab Halaman Inject Star



Gambar 3.6 Halaman tab inject star struk toko

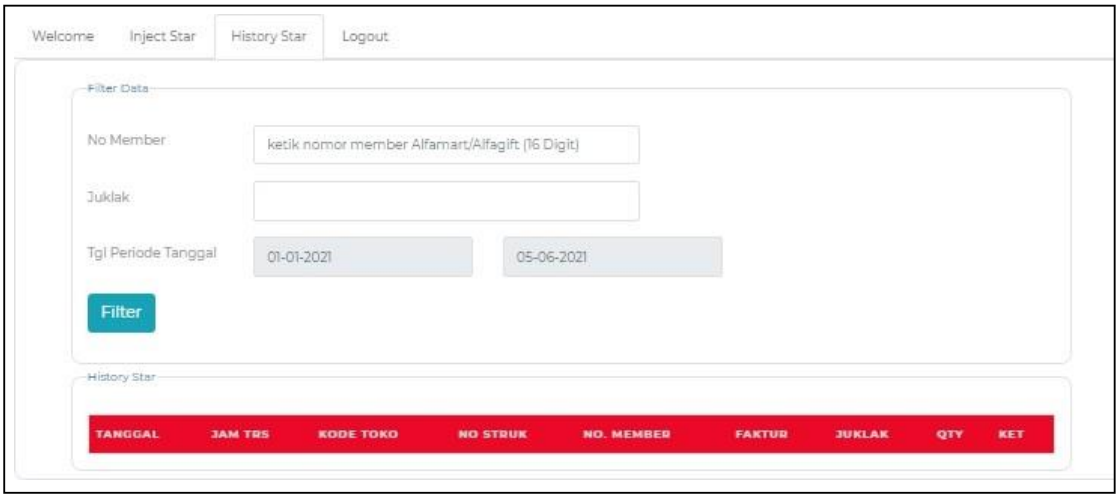
Pada Gambar 3.6 bisa dilihat tampilan halaman inject star struk toko yang berisikan no member, kode toko, nomor struk, tanggal transaksi, jam transaksi, faktur, nomor juklak dan *quantity* dan juga ada tombol simpan.



Gambar 3.7 Halaman tab inject star struk alfagift

Pada Gambar 3.7 menunjukan halaman inject star struk alfagift yang berisikan nomor member, no order, tanggal transaksi, jam transaksi, faktur, no juklak dan *quantity*.

B.3.3 Tab Halaman History Star



Gambar 3.8 Tab halaman history star

Pada Gambar 3.8 menunjukan tampilan di *tab* halaman history star yang berisikan no member, juklak, dua *range* tanggal periode dan tombol filter.

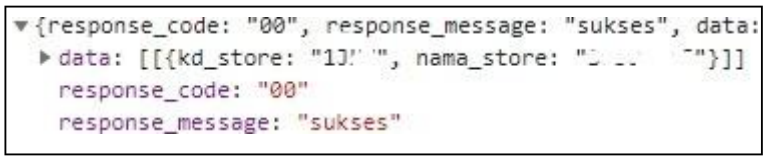
C. Implementasi

Berdasarkan rancangan aplikasi *flowchart*, struktur API dan rancangan *user interface*. Berikut adalah implementasinya.

C.1. Implementasi API

Berikut adalah implementasi API yang telah dibuat dalam pengerjaan aplikasi alfastar.

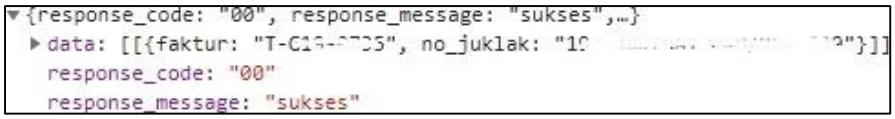
C.1.1 GET KodeToko



Gambar 3.9 Response message kode toko

Pada Gambar 3.9 merupakan *response message* kode toko di halaman *tab* inject star struk toko yang berasal dari API *method get* kode toko.

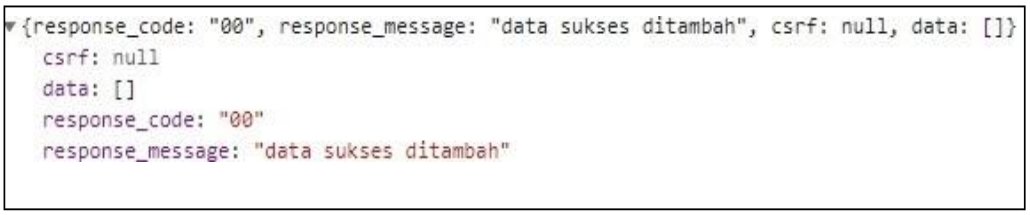
C.1.2 GET Juklak



Gambar 3.10 Response message juklak

Pada Gambar 3.10 merupakan *response message* input juklak pada di halaman inject star struk toko dan inject star struk alfa gift yang berasal dari API *method* GET juklak.

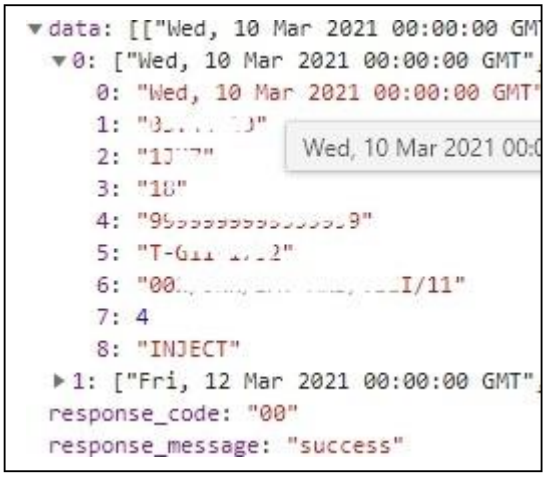
C.1.3 Inject Star



Gambar 3.11 Response message inject star

Pada Gambar 3.11 merupakan *response message* saat mengisi star pada di halaman inject star struk toko dan inject star struk alfa gift yang berasal dari API *method post* star.

C.1.4 History Star



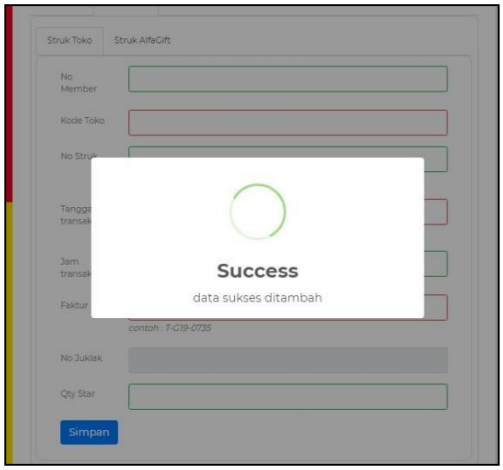
Gambar 3.12 Response message history star

Pada Gambar 3.13 merupakan *response message* saat menjalankan API history star yang menggunakan metode *get*.

C.2 Implementasi Aplikasi

Berikut adalah implementasi aplikasi alfastar yang sudah dibuat berupa tampilan antarmuka dalam berbentuk halaman web.

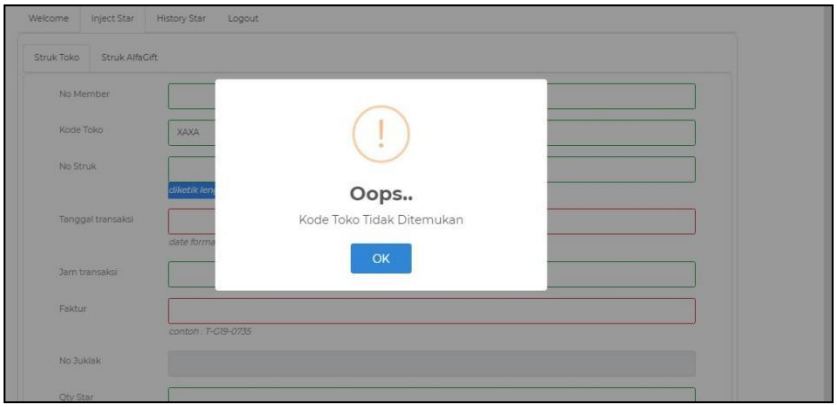
C.2.1. Implementasi Halaman Inject Star



Gambar 3.13 Hasil saat inject star

Pada Gambar 3.13 diperlihatkan *output* jika berhasil menambahkan atau melakukan proses inject star / API inject.

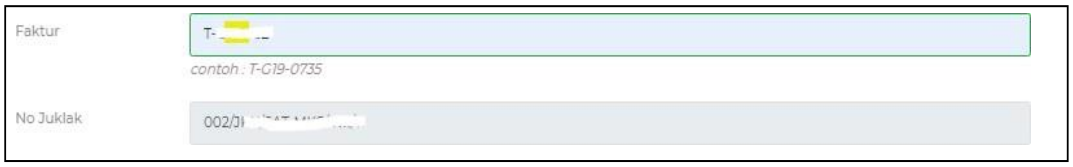
C.2.2 Implementasi Kode Toko



Gambar 3.14 Output dari input kode toko

Pada Gambar 3.14 diperlihatkan *output* jika tidak berhasil menemukan kode toko saat ingin melakukan inject star dan ditemukan saat proses *get* kode toko.

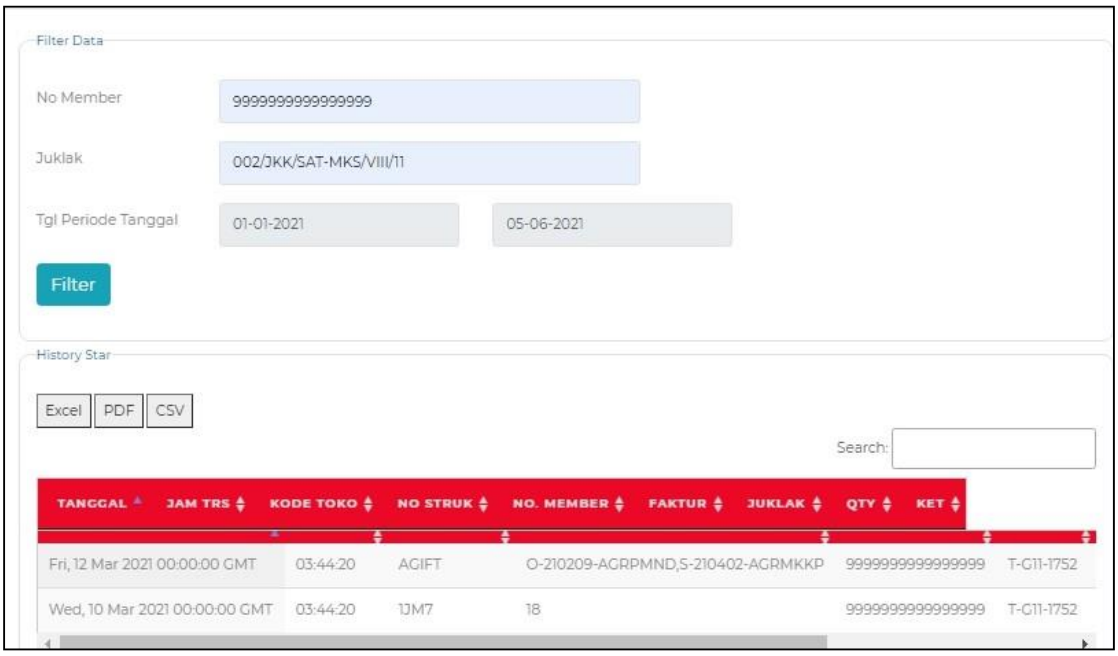
C.2.3 Implementasi Juklak



Gambar 3.15 Output dari input faktur

Pada Gambar 3.15 adalah hasil dari *input* faktur untuk mengambil nomor juklak yang diambil menggunakan API.

C.2.4 Implementasi History Star



Gambar 3.16 Implementasi history star

Pada Gambar 3.16 menunjukan implementasi history star jika berhasil menemukan data yang didapat dari API history star.

Gambar 3.17 Tidak ada data history

Pada Gambar 3.17 adalah hasil dari masukkan akan tetapi tidak ada record di dalam database.

3.3.2 Kendala yang Ditemukan

Berikut adalah kendala yang ditemukan saat pengerjaan aplikasi saat kerjamagang di PT. Global Loyalty Indonesia

1. Pengerjaan API dilakukan secara bersamaan dengan pembuatan *front-end* sehingga pengerjaan cenderung lambat.
2. Pengerjaan API dan *front-end* cenderung lambat dikarenakan penggunaan framework baru dipelajari saat kerja magang.
3. Adanya bugs saat melakukan inject dan pemanggilan history yaitu *logout* dari sistem sendiri.

3.3.3 Solusi Atas Kendala Yang Ditemukan

Berikut adalah solusi yang dilakukan saat pengerjaan aplikasi saat kerja magang di PT. Global Loyaty Indonesia.

1. Melakukan *debugging* dengan cara memasukkan breakpoint saat API dijalankan.
2. Bertanya kepada Senior di tempat jika mengalami kesulitan.
3. Belajar dari website seperti tutorialpoints jika menemukan *error*.